IChemE

P.D. 10/1997 P.693-708 16

XP-001023928

0263-8762/97/$10.00+0.00

ROSSMPC: A NEW WAY OF REPRESENTING AND ANALYSING PREDICTIVE CONTROLLERS

M. TVRZSKÁ DE GOUVÊA and D. ODLOAK

Chemical Engineering Department, University of São Paulo, São Paulo, Brazil

There is an increasing tendency towards the use of a lesser number of controllers in industry. This is so as the inclusion of an economical optimization layer becomes simpler. However, a reduced number of controllers means that they are large ones. These must be adequately tuned, which can be accomplished through their state space representation. Several ideas for obtaining a state space representation of the class of model predictive controllers (MPC) have appeared in the literature like the approach of Li *et al.*[1]. Nevertheless, most of them lead to a system with thousands of states if the controller becomes large. As a consequence, the computation of eigenvalues or singular values to evaluate the stability of the system becomes prohibitive. In this paper a new state space representation of the MPC controllers is introduced based on the continuous step response of the system given by a transfer function. The idea is to represent the step response in a parametric form. The number of states of this representation is of the same order as the identified transfer function however large the optimization horizon of the chosen predictive controller might be. The method includes integrating and open loop unstable processes. Besides that the model time delay appears explicitly and is independent of other parameters. The computation work involved in the closed loop analysis is thus minimized. Also, this new representation can be used to robustly tune the MPC controllers without the use of extensive simulations.

Keywords: model predictive controllers; robust control

I. INTRODUCTION

For almost two decades, linear model predictive controllers have been extensively used in industry (Richatet *et al.*[2], Cutler and Ramaker[3], Prett and Gillette[4]). The main reason for their acceptance is their capability of enforcing constraints on variables and their ability to capture the economic benefits by an optimal operation of the plant. However, two additional interesting properties of this type of controller should not be forgotten, namely, their stabilizing properties and their multivariable character.

Both of these are important for achieving optimal operation of a plant. In recent times the demand for plant optimization has grown enormously. The plant economic objectives are more easily defined for the full unit. At the same time it is easier to implement the optimization commands (normally given as set-points to the controllers) if there are a small number of controllers. The ideal situation would be to have a single controller for the whole plant. If one considers a real process, e.g. if the optimization of a crude distillation unit or a fluid catalytic cracking unit is to be undertaken, the number of manipulated inputs may amount to fifty and the number of controlled outputs may be near a hundred. From the implementation point of view there are also advantages in increasing the size of the controllers. The controller commissioning gains more flexibility since this work can be done by switching on the controller per section of the plant.

Thus it can be seen that being able to deal with a large controller is what should be aimed for. On the other hand, most of the industrial MPC packages such as DMC (Cutler and Ramaker[3]), MPHC (Richalet *et al.*[2]), MAC (Rouhani and Mehra[5]) and STAR (Guettler *et al.*[6]) still use the same concepts of the early predictive controllers, i.e. they are based on the step response model of the process. Therefore, the number of coefficients in the models used by the controller may be greater than 100. This is necessary if a good performance of the system is required.

The MPC packages when applied to a large number of inputs and outputs may have to deal with some thousands of parameters and model coefficients. With the available computer power and optimization software, this might not be a major problem for the calculation of the control actions. However, as far as the tuning of the controller is concerned, problems may arise. The work involved in the usual practice of simulating the system for typical disturbances and model uncertainties tends to become overwhelming, as the number of possible situations will increase at an exponential rate. Frequently, analysis tools cannot be found for the design procedure of the MPC. The availability of these can, however, simplify this procedure.

Li *et al.*[1] showed how to represent the usual unconstrained MPC strategy in the state space representation. By doing so the way to the closed loop analysis of the predictive controllers was open. This analysis encompasses many features, as for example:

- how to evaluate the robustness of the controller: without extensive simulation it is possible to evaluate the performance of the controller submitted to model mismatch

or how to select its parameters so that the performance remains acceptable for a wide family of identified models.
• how to evaluate whether the controller maintains a suitable response, e.g. by constraining the closed loop poles inside a closed region of the complex plane.

So, if the system is in the state space form, an extensive library of methods and results related to the linear system theory could be incorporated as analysis tools for the model predictive (MP) controllers. The available procedure (Li *et al.*[1] and Lundstrom *et al.*[7]), nonetheless, produces for a large system a model with thousands of states as each step coefficient corresponds to a new state. Under this scenario there is a clear incentive to obtain a more economic representation of the system!

In this paper a more compact state space representation of the class of MP controllers (named ROSSMPC) that uses step response based internal models is presented. In section II the existing approaches is briefly reviewed and it is shown why they may become of little use for an extensive closed loop robust analysis. In section III a state space representation is presented and the way the order of the model can be reduced is shown. In section IV some examples are presented where it is shown how ROSSMPC can be used in the analysis and design of MPC controllers. For example it will be shown that the DMC controller can have the same performance as the OBMPC controller if properly designed, contrary to what was claimed in Lundstrom *et al.*[7]. In section IV it is also shown that a subsystem of the well known Shell problem (Prett and Garcia[10]) can be stabilized with the present ROSSMPC formulation with a much lesser optimization horizon than the ones obtained with the conventional representations presented in the literature.

As the controller must retain its good performance in the real environment, it is further shown how a new analysis tool (Hudchison and Pritchard[11] and Li *et al.*[12]) can be used to quantify the robustness of the chosen controller. This tool is generalized for some structured systems. This is shown in section V where this method is introduced as an additional tool that can be used together with the existing techniques for the design of MPC. In section VI some examples are presented of how these tools can be used and the paper finally concludes in section VII.

II. THE EXISTING STATE SPACE REPRESENTATIONS AND THEIR LIMITATIONS

Most of the original MP controllers make use of step or pulse response models to predict the system output. The coefficients of these models can be obtained by their fitting to plant data (Bailey[13]). The model so obtained is usually of a very high order. Time delay is accommodated by placing zeros in the first model coefficients.

One point that must be emphasized is how the time delay is enabled to vary in this model. This is due to the fact that in the closed loop analysis the effect of the time delay on the stability or performance of the system must be considered. If it appears interrelated with other model parameters, the analysis becomes more complicated. In the step response representation changes, in the time delay can be considered in the following two ways:

(a) The new time delay is equal to the old time delay plus or minus a multiple of the sampling time. In this case the new model is obtained simply by adding or removing leading zeros from the old model. The remaining model coefficients are not modified. Therefore the time delay is an explicit parameter.
(b) The new time delay affects the whole response. In this case not only the leading zeros are affected but also the coefficients of the step or pulse response model. Consequently it can be considered that in this model formulation, the time delay is not an independent or explicit parameter of the model.

Not only is the time delay an important parameter that should be made explicit whenever possible, there are other intuitive parameters of the system that should be made explicit. For example the control engineer has a good idea of how much the gains and time constants of the real process can change in the real system. This notion is very important in the closed loop analysis. Though it is not simple to associate these instincts with the coefficients in the step or pulse response models.

With the introduction of the Matlab Identification Toolbox® another model became popular, namely the pulse transfer function (Ljung[14]). This is usually a low order parametric model where all the intuitive aforementioned parameters always appear explicitly except the time delay.

Lee and Yu[19] and Oliveira and Biegler[20] propose a general state space structure to represent and analyse the MP controllers. In their representation, time delays are not considered and they give no indications on how to deal with the situation when this parameter can vary continuously. Another limitation of their approach is that the resulting relation between the predictions of the system output and the model parameters is nonlinear. This last aspect complicates the robustness analysis of the closed loop.

Most chemical engineering processes are continuous and can be adequately represented by continuous transfer functions. For MIMO systems the relation between any controlled output $y_i \in IR^{n_y}$ and any manipulated input $u_j \in IR^{n_u}$ may be represented by equation (1).

$$G_{i,j}(s) = \frac{B_{i,j}(s)e^{-s\theta_{i,j}}}{A_{i,j}(s)} \tag{1}$$

where

$G_{i,j}$	is the process continuous transfer function
i,j	are indexes that are related to the controlled and manipulated variables
s	is the Laplace domain
$\theta_{i,j}$	is the time delay appearing in an explicit manner
$B_{i,j}$ and $A_{i,j}$	are polynomials given by:

$$B_{i,j}(s) = b_{i,j,0} + b_{i,j,1}s + b_{i,j,2}s^2 \ldots + b_{i,j,nb_{i,j}}s^{nb_{i,j}}$$

$$A_{i,j}(s) = 1 + a_{i,j,1}s + a_{i,j,2}s^2 \ldots + a_{i,j,na_{i,j}}s^{na_{i,j}}$$

$b_{i,j,k}$ and $a_{i,j,k}$ are the coefficients of the model
$nb_{i,j}$ and $na_{i,j}$ are the orders of the polynomials

The polynomial $A_{i,j}(s)$ is usually assumed to have the single roots: $r_{i,j,1}$, $r_{i,j,2}$, ..., $r_{i,j,na_{ij}}$. This is not a serious restrictive assumption, since poles of high multiplicity are not frequently found in the process industry. The roots of the polynomial $A_{i,j}(s)$ correspond to the inverses of the time

constants of the system. The process gain and the parameters of eventual inverse responses are directly associated with the coefficients of the numerator of the transfer function, namely $B_{i,j}(s)$.

So it can be seen that the model of equation (1) seems to be an adequate model that could be used in a closed loop analysis of MP controllers. As the implementation is normally carried with the convolution model, the relationship between these two formulations must be obtained, and this is presented next.

If a step is applied to the system described by equation (1), the corresponding step response model can be obtained for any instant as given in equation (2).

$$s_{i,j,n} = \sum_{g=0}^{na_{i,j}} c_{i,j,g} f_{i,j,g}(nT) \quad (2)$$

and the sampling instant is such that $nT > \theta_{i,j}$. The coefficients $c_{i,j,g}$ are the residues of $G_{i,j}/s$ and $f_{i,j,g}$ is given by:

$$\text{for } g = 0: \quad f_{i,j,0}(t) = 1$$

$$\text{for } g \neq 0: \quad f_{i,j,g}(t) = \begin{cases} e^{r_{i,j,g}(t-\theta_{i,j})} : r_{i,j,g} \neq 0 \\ t - \theta_{i,j} : r_{i,j,g} = 0 \end{cases} \quad (3)$$

From the above equations it is clear that the step response coefficients are explicit functions of the parameters of the transfer function model. The gain of the system relating input u_j and output y_i is related linearly with the coefficients $c_{i,j,g}$. So if there is any uncertainty in the parameters of the model (due to e.g. plant mismatch) its influence can be easily evaluated. For instance, an increase of a given percentage in the plant gain corresponds to the same percentage increase in $c_{i,j,g}$ for all g.

The Closed Loop Formulation of OBMPC

For open loop stable processes, the state space model presented by Li *et al.*[1], allows the state predictor of MPC to be written as:

$$[y]_{k+1/k} = M[y]_{k/k} + S\Delta u_k \quad (4)$$

$$[y]_{k+1/k} = [y]_{k/k} + R\{\hat{y}_k - N[y]_{k+1/k}\} \quad (5)$$

where

$$[y]_{k/k} = \left[y_{k+1}^T y_{k+2}^T \ldots y_{k+n_h}^T\right]_{k/k}^T \quad (6)$$

$$\Delta u_k = u_k - u_{k-1}$$

$$M = \begin{bmatrix} 0 & I_{n_y} & 0 & \ldots & 0 & 0 \\ 0 & 0 & I_{n_y} & \ddots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \ddots & I_{n_y} & 0 \\ 0 & 0 & 0 & \ddots & 0 & I_{n_y} \\ 0 & 0 & 0 & \ldots & 0 & I_{n_y} \end{bmatrix}_{(n_y \cdot n_h)} \quad S = \begin{bmatrix} S_2 \\ S_3 \\ \vdots \\ S_{n_h-1} \\ S_{n_h} \\ S_{n_h+1} \end{bmatrix} \quad (7)$$

S_i is the i^{th} coefficient of the step response

n_h is the number of coefficients of the step reponse model

R is the plant output correction gain. It can be easily calculated following the procedure of the Kalman filter[15]. In this case Lundstrom *et al.*[7] designate the controller OBMPC (Observer Based MPC)

$\hat{y}_k$ is the real measured output vector at instant k

N is a matrix which collects the first n_y components of the state vector

$$N = \left[\underbrace{I_{n_y}\ 0 \ldots 0}_{n_h}\right]$$

If a model with the same structure of the predictor but with different parameters is used to represent the real plant in the closed loop analysis, a total of $2n_y.n_h$ states will be necessary. Since n_h is usually about 100 and n_y may go up to 50 for a large controller, the method would generate near *ten thousand states*. This is certainly not suitable for practical purposes.

III. THE REDUCED ORDER STATE SPACE MODEL PREDICTIVE CONTROLLER (ROSSMPC)

In the previous section it was shown how a state space representation of the predictive controllers based on step responses curves can be obtained. The approach shown there was that of Li *et al.*[1] It could be seen that the number of states was very large. Moreover, it was equal to the order of the convolution model, i.e. equal to the number of coefficients. So the number of states does not depend on two interesting tuning parameters of the controller, namely the control and optimization horizons. One attempt to reduce the number of states was made by Hovd *et al.*[9] and Lundstrom *et al.*[7] The authors have noticed that a truncation of the step response curve will reduce the number of states. In order to preserve the information of the truncated part, a parametric first order model was chosen to approximate the tail of the curve. This approach, besides being approximate and consequently not useful for closed loop analysis, is only valid for open loop stable processes.

In regard to the aspects considered in the last paragraph, the points that must be taken into account for obtaining a good representation of the controller can be summarized as follows:

- as the optimization horizon can be much smaller than the order of the convolution or step response model, it would be desirable to have a representation that would be dependent on the optimization horizon.
- the formulation of the controller should be valid for both stable and unstable open-loop processes.

In addition to these aspects, another point can be outlined. Rowlings and Muske[16] showed that under the hypothesis of ideal model and open loop stable systems, the MPC is intrinsically stable for any tuning parameters if the optimization horizon tends to infinity. So it seems reasonable that for the real process where there is model uncertainty, a long prediction horizon tends to stabilize the system. This justifies a practical rule that recommends the use of an optimization horizon equal to the open loop stabilizing time of the system. With this rule there are processes for which the MPC requires a large control horizon. That means that even if the number of states were reduced according to the criterion based on the optimization

horizon, this would be not acceptable. So another desirable characteristic of the representation would be:

- the capability of reducing the number of states when the optimization horizon must be large for stabilizing purposes

Our approach, named ROSSMPC (reduced order state space model predictive controller) incorporates these three desired characteristics. In the rest of this section the ROSSMPC formulation will be introduced. This will start with the prediction model that is dependent on the optimization horizon rather than on the convolution order. So it will be shown how to transform the dependency on the convolution order into a dependency on the optimization horizon. As this will be done it will also be shown that the approach is also valid for unstable processes. After this the closed loop response is derived together with the control action. Finally it will be shown how to reduce the obtained representation making use of different fictitious sampling periods.

Building the Prediction Model of the ROSSMPC

The starting point will be a function whose output is the prediction curve of the controlled variables given by a weighted composition of the step responses in the inputs. It will be obtained on all available information up to the instant k. So the ROSSMPC prediction model has the general form given by:

$$[y_{i,k+n}]_{k/k} = [\bar{p}_i]_{k/k} + \sum_{j=1}^{n_u} \sum_{g=1}^{na_{i,j}} [p_{i,j,g}]_{k/k} f_{i,j,g}(nT) \tag{8}$$

where, $i=1, \ldots, n_y$, $j=1, \ldots, n_u$, $g=1, \ldots, na_{i,j}$, $\bar{p}_i$ and $p_{i,j,g}$ are the new states of the model that will be called artificial states in the remaining of the paper and n is taken such that $nT > \max_j \theta_{i,j}$. In multivariable form equation (8) becomes

$$[y_{k+n}]_{k/k} = [\bar{P}]_{k,k} + \psi [P]_{k,k} \tag{9}$$

where,

$$[y_{k+n}] = \left[y_{1,k+n} y_{2,k+n} \cdots y_{n_y,k+n}\right]^T$$

$$\bar{P} = \left[\bar{p}_1, \bar{p}_2 \cdots \bar{p}_{n_y}\right]^T$$

$$P = \left[p_{1,1,1}\; p_{1,1,2} \cdots p_{1,1,na_{1,1}}\; p_{1,2,1}\; p_{1,2,2} \cdots p_{1,2,na_{1,2}} \cdots p_{n_y,1,1}\; p_{n_y,1,2} \cdots p_{na_{ny,1}} \cdots p_{n_y,n_u,1}\; p_{n_y,n_u,2} \cdots p_{n_y,n_u,na_{ny,nu}}\right]^T$$

$$\psi = \left[\begin{array}{ccccccccccccccc} f_{1,1,1} & f_{1,1,2} & \cdots & f_{1,1,na_{1,1}} & f_{1,2,1} & \cdots & f_{1,2,na_{1,2}} & \cdots & f_{1,n_u,na_{1,nu}} & 0 & 0 & \ldots & 0 & 0 & \ldots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & \ldots & 0 & f_{2,1,1} & f_{2,1,2} & \cdots & f_{2,1,na_{2,1}} & f_{2,2,1} & \cdots \\ \vdots & \vdots & \ldots & \vdots & \vdots & \ldots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \ldots \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & \ldots \end{array}\right.$$
$$\left.\begin{array}{cccccccccccccccccccc} \ldots & 0 & \ldots & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 \\ \ldots & f_{2,2,na_{2,2}} & \ldots & f_{2,n_u,1} & \ldots & f_{2,n_u,na_{2,nu}} & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & \ldots & 0 \\ \ldots & \vdots & \ldots & \vdots & \ldots & \vdots & \vdots & \ldots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots & \vdots & \ldots & \vdots \\ \ldots & 0 & \ldots & 0 & \ldots & 0 & 0 & \ldots & f_{n_y,1,1} & f_{n_y,1,2} & f_{n_y,1,na_{ny,1}} & f_{n_y,2,1} & \cdots & f_{n_y,2,na_{ny,2}} & \cdots & f_{n_y,n_u,1} & \cdots & f_{n_y,n_u,na_{ny,nu}} \end{array}\right] \tag{10}$$

In equation (10) ψ is calculated in the instant (nT).

Next, how the artificial states of this model can be obtained will be discussed. Note that equation (8) is very similar in form equation (2). This enables the artificial states to be updated as follows:

$$[\bar{p}_i]_{k+1/k} = [\bar{p}_i]_{k/k} \sum_{j=1}^{n_u} c_{i,j,0} \Delta u_{j,k} \tag{11}$$

$$[p_{i,j,g}]_{k+1/k} = d_{i,j,g} [p_{i,j,g}]_{k/k} + c_{i,j,g} d_{i,j,g} \Delta u_{j,k} \tag{12}$$

where

$$d_{i,j,g} = e^{r_{i,j,g} \cdot T} \quad \text{if} \quad r_{i,j,g} \neq 0$$

$$d_{i,j,g} = \frac{(t + T - \theta_{i,j})}{t - \theta_{i,j}} \quad \text{if} \quad r_{i,j,g} = 0$$

Or alternatively in compact multivariable form as:

$$[\bar{P}]_{k+1/k} = [\bar{P}]_{k/k} + C_0 \Delta u_k \tag{13}$$

$$[P]_{k+1/k} = D[P]_{k/k} + CD\Delta u_k \tag{14}$$

where,

$$C_0 = \begin{bmatrix} c_{1,1,0} & c_{1,2,0} & \cdots & c_{1,n_u,0} \\ \vdots & \vdots & \cdots & \vdots \\ c_{n_y,1,0} & c_{n_y,2,0} & \cdots & c_{n_y,n_u,0} \end{bmatrix}$$

$$C = diag\big(c_{1,1,1}\; c_{1,1,2} \ldots c_{1,1,na_{1,1}}\; c_{1,2,1}\; c_{1,2,2} \cdots c_{1,2,na_{1,2}} \ldots c_{1,n_u,1} \cdots c_{1,n_u,na_{1,nu}} \cdots c_{2,1,1} \cdots c_{2,n_u,na_{2,nu}} \ldots c_{n_y,1,1}\; c_{n_y,1,2} \cdots c_{n_y,n_u,na_{ny,nu}}\big)$$

$$D = diag\big(d_{1,1,1}\; d_{1,1,2} \ldots d_{1,1,na_{1,1}}\; d_{1,2,1}\; d_{1,2,2} \cdots d_{1,2,na_{1,2}} \ldots d_{1,n_u,1} \cdots d_{1,n_u,na_{1,nu}} \cdots d_{2,1,1} \ldots d_{2,n_u,na_{2,nu}} \ldots d_{n_y,1,1}\; d_{n_y,1,2} \ldots d_{n_y,n_u,na_{ny,nu}}\big)$$

As low order transfer function models are normally used to represent the system, the number of states generated by equations (13) and (14) are small. This means that the reduction in the number of states when considering equations (13) and (14) instead of the states corresponding to the coefficients of the step response ranging from the optimization horizon (n_r) up to the stabilizing horizon (n_h) is substantial. An additional advantage of this model representation is that the prediction of the system output can be arbitrarily extrapolated including time instants far into the future without significantly increasing the order of the state space model that represents the closed loop.

It is not a common feature of the MPC packages found in industry to consider not equally spaced prediction time instants. However, with the proposed model structure a

controller can be designed with a large optimization time horizon and few prediction instants which results in a small closed loop state space model. This property will be commented on later.

Equation (4) can also be used as the prediction model of ROSSMPC. The only difference lies in the definition of the matrices M, S and the vector of states. This last is redefined so as to incorporate the artificial states of the model in P and $\bar{P}$. The new matrices M and S are obtained by combining equations (4), (13) and(14). Thus:

$$[y]_{k+1/k} = M[y]_{k/k} + S\Delta u_k \tag{4}$$

with

$$y = \left[y_{k+1}^T y_{k+2}^T \ldots y_{k+n_r}^T \bar{P}^T P^T\right]^T$$

$$M = \begin{bmatrix} 0 & I_{n_y} & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & I_{n_y} & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & I_{n_y} & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & I_{n_y} & \psi((n_r+1)T) \\ 0 & 0 & 0 & \ldots & 0 & I_{n_y} & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & D \end{bmatrix}_{[n_y(n_r+1)+\sum_{ij} na_{ij}]}$$

$$S = \begin{bmatrix} S_2 \\ S_3 \\ \vdots \\ S_{n_r} \\ S_{n_r+1} \\ C_0 \\ CDN_p \end{bmatrix} \qquad S_i = \begin{bmatrix} s_{1,1,i} & s_{1,2,i} & \ldots & s_{1,n_u,i} \\ s_{2,1,i} & s_{2,2,i} & \ldots & s_{2,n_u,i} \\ \vdots & \vdots & \ldots & \vdots \\ s_{n_y,1,i} & s_{n_y,2,i} & \ldots & s_{n_y,n_u,i} \end{bmatrix} \tag{15}$$

Equation (4) is used to predict what will happen if no other control actions are taken. As the process is to be stabilized, it must be known what the states will be like when steady-state is reached. Here lies the main difference between our approach and others. For the conventional MPC strategies (e.g. DMC, MAC, etc) there is the need to predict n_h future instants, where n_h is the time needed for the process to stabilize. For real processes this number is usually very large (100 or larger). In the ROSSMPC strategy presented so far there is only the need to predict $n_r n_y + n_p$ future values, where n_r is the optimization horizon and n_p is the number of artificial states in P and $\bar{P}$ and can be obtained as:

$$n_p = n_y + \sum_{i=1}^{n_y}\sum_{j=1}^{n_u} nd_{ij}$$

For real processes the required number n_r may be as low as 10 and a typical value of na_{ij} is 2. Also, for a real process n_h is of the order of 100. Using the ROSSMPC representation for a SISO system, only need 13 states would therefore be needed in equation (4), while the conventional state-space representations of the MPC's will need n_h states, i.e. about 100 states. So it can be seen that there is a significant reduction in the number of states required.

Before going further, comments are needed on some characteristics of our model. Observing equation (15) and the definition of matrix ψ (equation (10)) it can be seen that all the terms in matrices M and S preserve a linear relation with each of the parameters of the original step response model (equation (2)). This is important if the uncertainties in the model are to be taken into account. Also their continuous form is preserved and this makes it possible to search for the worst case condition where the controller might have performance or stability problems as will be seen later in this paper.

Again, it is desirable to correct the prediction with measured data which is accomplished by the use of equation (5). Combining equations (4) and (5) the prediction model given in equation (16) is finally achieved.

$$[y]_{k+1/k+1} = (I - RN)M[y]_{k/k} + R\hat{y}_k + (I - RN)S\Delta u_k \tag{16}$$

The control actions are calculated so as to minimize an objective function. MP controllers assume that equation (17) is the objective function. In equation (17) the output is predicted in n_r future instants and m control actions are used to drive the process to the desired states. The effect of the future control actions are predicted according to equation (18).

$$J = \min_{\Delta U_k}\left\{\left\|\Gamma(y^{sp} - [y]_{k/k}^p)\right\|^2 + \|\Lambda\Delta U_k\|^2\right\} \tag{17}$$

$$[y]_{k/k}^p = [y]_{k/k} + \bar{S}\Delta U_k \tag{18}$$

where,

Λ,Γ	are diagonal weighting matrices
$y^{sp} \in IR^{n_r \times n_y + n_p}$	are the desired values for the states and will be commented on later.
ΔU_k	are the future calculated control actions given by:

$$\Delta U_k = \left[\Delta u_k^T \Delta u_{k+1}^T \ldots \Delta u_{k+m-1}^T\right]^T$$

$$\bar{S} = \begin{bmatrix} S_1 & 0 & 0 & \ldots & 0 \\ S_2 & S_1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ S_m & S_{m-1} & \ldots & \ldots & S_1 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ S_{n_r} & S_{n_r-1} & \ldots & \ldots & S_{n_r-m+1} \end{bmatrix}$$

and S_i are the same as in equation (15).

The well known least-squares solution of equation (17) with the prediction (18) is presented in equation (19).

$$\Delta U_k = \left\{\bar{S}^T\Gamma^T\Gamma\bar{S} + \Lambda^T\Lambda\right\}^{-1}\bar{S}^T\Gamma^T\Gamma\left\{y^{sp} - [y]_{k/k}\right\} \tag{19}$$

As only the first control action is of interest, the following can be we obtained:

$$\Delta u_k = \left[I_{n_u}\ 0 \ldots 0\right]\Delta U_k$$

So equation (19) can be written in a compact form as equation (20).

$$\Delta u_k = K_{MPC}\left(y^{sp} - [y]_{k/k}\right) \tag{20}$$

A natural question that can arise is how to evaluate y^{sp} for the artificial states in P and $\bar{P}$. To answer this question, note that it is necessary to observe that if equation (4) is to be valid at steady state, P has to be driven to zero and $\bar{P}$ to the system output. Consequently, the components of y^{sp}

corresponding to P in equation (20) are set equal to zero while the components of y^{sp} corresponding to $\bar{P}$, have the same values as the set points of the outputs.

The Closed Loop Response of the ROSSMPC

Taking advantage from the fact that the last $n_p - n_y$ terms in y^{sp} must be zero and the particular structure of matrix M, the following relation can be obtained from equations (16) and (20):

$$\{y^{sp} - [y]_{k+1/k+1}\} = (I - RN)(M - SK_{MPC})\{y^{sp} - [y]_{k/k}\} + RN\{y^{sp} - [\hat{y}]_{k/k}\} \qquad (21)$$

As the model used by the controller can differ from the real plant, this should be taken into account for obtaining the closed loop transfer function of the process. The real plant is assumed to follow the equations given below:

$$[\hat{y}]_{k+1/k+1} = \hat{M}[\hat{y}]_{k/k} + \hat{S}\Delta u_k \qquad (22)$$

where $\hat{M}$ and $\hat{S}$ are similar in form to M and S in equation (15).

Note that for the real process there is no need to predict n_r future instants. It is only necessary to evaluate $\hat{n}_r$ future instants. Where $\hat{n}_r$ is such that:

$$\hat{n}_r T > \max_{i,j} \hat{\theta}_{i,j}$$

So if the control actions are taken into account the relation given in equation (23) and the closed loop transfer function of equation (24) can be obtained.

$$\{y^{sp} - [y]_{k+1/k+1}\} = [(I - RN)(M - SK_{MPC}) - RN\hat{S}K_{MPC}]\{y^{sp} - [y]_{k/k}\} + RN\hat{M}\{y^{sp} - [\hat{y}]_{k/k}\} \qquad (23)$$

$$\begin{bmatrix} y^{sp} - [y]_{k+1/k+1} \\ y^{sp} - [\hat{y}]_{k+1/k+1} \end{bmatrix} = \begin{bmatrix} H & RN\hat{M} \\ -\hat{S}K_{MPC} & \hat{M} \end{bmatrix} \begin{bmatrix} y^{sp} - [y]_{k/k} \\ y^{sp} - [\hat{y}]_{k/k} \end{bmatrix} \qquad (24)$$

where

$$H = (I - RN)(M - SK_{MPC}) - RN\hat{S}K_{MPC}$$

Disturbances can be easily incorporated in equation (24). This is made so as to evaluate the performance of the control strategy. One popular approach often used in the literature is to consider an input disturbance in the plant as was done by Morari and Lee[16]. So, finally:

$$\begin{bmatrix} y^{sp} - [y]_{k+1/k+1} \\ y^{sp} - \hat{y}_{k+1} \end{bmatrix} = \begin{bmatrix} H & RN\hat{M} \\ -\hat{S}K_{MPC} & \hat{M} \end{bmatrix} \begin{bmatrix} y^{sp} - [y]_{k/k} \\ y^{sp} - \hat{y}_k \end{bmatrix} - \begin{bmatrix} 0 \\ \hat{S} \end{bmatrix} [du_k] \qquad (25)$$

where du_k is the input incremental disturbance.

The transition matrix defined by equation (24) or (25) describes the closed loop poles and consequently the stability of the system. Also the eigenvectors of this matrix define the directions where the system is more sensitive to disturbances or where the error in the predicted output has a slower decay. Therefore, it is critical that the dimension of this matrix be reduced to the smallest possible value so that it is possible to perform an intensive verification of the system behaviour under different conditions. These should be taken not only in terms of changes in the parameters of the plant, but also in terms of the tuning of the controller. The location of the poles of the transition matrix defines clearly the behaviour of the system. This practically eliminates the need of extensive simulation of the process, as will be shown in section IV.

Reducing the Order of the ROSSMPC—Final Formulation

If it is wished to reduce the order of the system and preserve the stabilizing property of the long range MP controllers, the prediction model must be slightly modified. In the usual approach the predicted values of the system output are calculated at equally spaced sampling instants along the optimization horizon. With the model representation of ROSSMPC the optimization horizon can be extended by including predictions of the system output at arbitrary future time instants. Or in other words a larger optimization horizon can be considered with fewer prediction points. As a consequence the state vector of the predictor becomes smaller.

In the conventional approach the predicted output is given by:

$$y_{k+j} \quad \text{where} \quad j = 1, 2, \ldots, n_h$$

In the extended approach it is considered that the prediction is carried in the following instants j:

$$j = 1, 2, \ldots, n_t, n_t + n_1, n_t + n_2, \ldots, n_t + n_k$$

where

$$n_t T > \max_{i,j} \theta_{i,j}$$

and $n_1, n_2, \ldots, n_k$ correspond to not necessarily equally spaced time instants and can be conveniently selected so as to take advantage of the stabilizing properties of the long prediction horizon. The idea of the method is to select a small horizon n_t, complemented by another small set of sampling points $n_1, n_2, \ldots, n_k$. The complete optimization horizon $n_k + n_t$ would be similar to the optimization horizon n_r of the conventional MP controllers. The state variables are so now defined as:

$$y = [y_{k+1} \; y_{k+2} \; \cdots \; y_{k+n_t} \; y_{k+n_t+n_1} \; y_{k+n_t+n_2} \; \cdots \; y_{k+n_t+n_k} \; \bar{P} \; P]^T$$

and the model coefficients of equation (15) are redefined as:

$$M = \begin{bmatrix} 0 & I_{n_y} & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & I_{n_y} & \ddots & 0 & 0 & \ldots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & I_{n_y} & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & I_{n_y} & \psi_{n_t} \\ 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & I_{n_y} & \psi_{Pn_t-n_j} \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & I_{n_y} & \psi_{n_t+n_k} \\ 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & I_{n_y} & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & D \end{bmatrix}$$

$$R = \begin{bmatrix} I_{n_y} \\ I_{n_y} \\ \vdots \\ I_{n_y} \end{bmatrix} \quad S = \begin{bmatrix} S_2 \\ S_3 \\ \vdots \\ S_{n_t} \\ S_{n_t+1} \\ S_{n_t+n_1+1} \\ \vdots \\ S_{n_t+n_k+1} \\ C_0 \\ CDN_p \end{bmatrix} \tag{26}$$

Observe that for the case of extrapolation $\psi_{n_t}, \psi_{n_t+n_1}, \ldots, \psi_{n_t+n_k}$ are calculated using equation (10) at the future prediction instants $(n_t+1)T, (n_t+n_1+1)T, \ldots, (n_t+n_k+1)T$.

Though it is not an usual approach to consider time instants that are not multiples of the sampling time, ROSSMPC can easily accommodate these, which can be particularly useful for interpolating the system output at time instants between the sampling instants. So this is another advantage of the ROSSMPC over the other existing MP representations.

IV. APPLYING THE ROSSMPC TO SEVERAL WELL KNOWN EXAMPLES

In the previous section the ROSSMPC representation was introduced. There it was shown how the existing linear MP controllers (DMC, MAC, OBMPC, etc) can be transformed into the ROSSMPC formulation. By doing so a significant reduction in the number of states can be obtained, allowing a better knowledge of what is really happening in the closed loop system. This is clearly shown in example 1. There it can be seen that using the ROSSMPC representation it can be understood why the performance of the DMC controller used by Lundstrom *et al.*[7] was poorer than the one of the OBMPC controller. Also the ROSSMPC representation can be used to better design the DMC controller and so similar performances will be obtained using either one of these controllers, which is shown in example 2. In example 3 the Shell problem is considered and it is shown how the ROSSMPC can be used to reduce the number of states of the system.

Example-1: ROSSMPC as an Analysis Tool

This example is intended to show how ROSSMPC can be used to analyse the performance of MP controllers. Here the DMC and the OBMPC controllers are considered.

The example is taken from Morari and Lee[17]. The process is given by the following transfer function and is assumed to be exactly known, i.e. no model uncertainty is considered in this example:

$$G(s) = \frac{100e^{-s}}{100s+1} \tag{27}$$

In order to achieve a high closed loop bandwidth, Morari and Lee[17] use a sampling time of 1min. As the steady-state is reached after about 300 min there is the need of about 300 coefficients in the step response model. That is, the approach of Li *et al.*[1] will need 300 states for representing the process model. Therefore, for the closed loop simulation altogether 600 states will be needed (300 for the process model and 300 for the real plant), which is obviously inconvenient for the purpose of design and analysis. The approach of Hovd *et al.*[9] would need at least 100 time steps for the truncated step response plus a first order system to approximate the tail of the step response. Thus, to represent the plant and model of the state predictor more than 200 states would be needed which is still excessive to represent this simple system. The number of states obtained by these two approaches is the same for the DMC and OBMPC. Using the ROSSMPC approach proposed in this paper a drastic reduction in the size of the state space model is obtained. For instance, for the DMC controller with a prediction horizon $n_r = 4$, the compact model is applied with only 11 states as follows: 4 states for model predictions, 2 states for model parameters, 3 states to calculate the plant output and 2 states as the new artificial states. Therefore, a reduction of over 90% in the number of states is obtained when the ROSSMPC representation is used. Note also that the response obtained by the ROSSMPC model is the same as the one obtained when using the conventional formulations of DMC, OBMPC, etc. as can be seen from the figures presented next.

As a compact step response model can be obtained, ROSSMPC may be used as an analysis tool to explain the

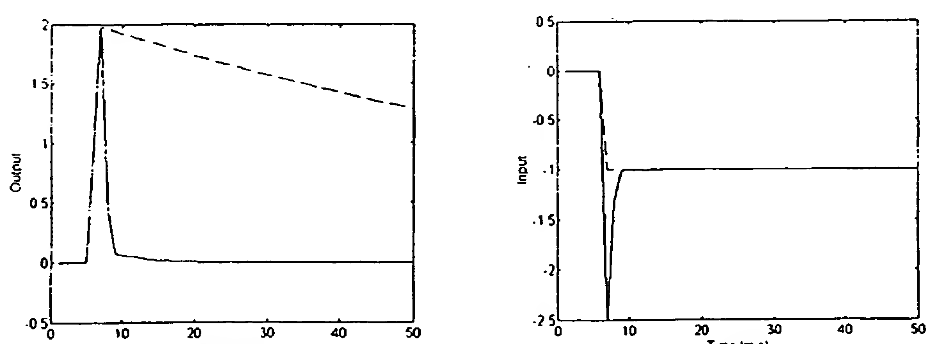

Figure 1. Responses for a step disturbance in the input: (- - -) DMC and (——) OBMPC.

performance of any controller. Next the approach for the DMC and OBMPC controllers will be exemplified, as these were used by Lundström *et al.*[7] It will also be considered that the process is subjected to input disturbance and the closed loop eigenvalues and eigenvectors of the system will be calculated for the ROSSMPC compact form. The DMC controller is simulated for the following parameters:

$$T = 1; \quad m = 2; \quad n_r = 4; \quad \hat{n}_r = 3; \quad \Lambda = 0$$

The disturbance is applied at the sampling time =5. Figure 1 shows the closed-loop responses, which are approximately the same ones obtained by Lundström *et al.*[7] From the figure, one might think that there is a clear superiority of OBMPC for this particular example. In fact in example 2 it will be seen that this may not be the case.

If in this case the eigenvalues of the transition matrix of equation (24) are inspected it can be observed that DMC drives the system poles to zero as a dead beat controller. The exception is a single pole that remains at 0.990 which corresponds to a time constant of 100 min for a sampling period of 1 min. In other words, the feedback strategy adopted by DMC for disturbance correction does not move the predictor pole from its open loop position. The normalized eigenvector that corresponds to this pole is:

$$\begin{aligned} \upsilon_{DMC} &= [y_{k+1} \ \cdots \ y_{k+4} \ \hat{p} \ p \ \hat{y}_{k+1} \ \hat{y}_{k+2} \ \hat{y}_{k+3} \ \hat{\hat{p}} \ \hat{p}] \\ &= [-.0141 \ -.0071 \ 0 \ .0070 \ .0731 \ -.7102 \\ &\quad -.0142 \ -.0141 \ -.0139 \ 0 \ -.0141]^T \end{aligned}$$

This vector defines the direction where the error calculated by equation (25) decays with a time constant equal to 100 min. This is exactly what happens for the output of DMC in Figure 1. It can be observed that components 1 to 4 and 9 to 11 of υ_{DMC}, corresponding respectively to model and plant states, are small compared to components 5 and 6 which correspond to the states of model parameters. This means that the slow approach of the error vector to zero is performed with small errors in the model and plant states.

One last comment is noteworthy. The apparent large error of DMC in Figure (1) is related to the extremely high gain of the system represented by equation (27). A better criterion of performance should be to compare the disturbance rejection of both approaches observing the open loop response of the system. For the input disturbance case, after 30 min in open-loop the system output would be about 25, while with DMC the system output is 1.5. This means that the controller rejects 94% of the disturbance effect, which is a reasonable performance for any practical standard. On the other hand, OBMPC rejects more than 99.9% of the same disturbance after 30min. This superior performance can be explained if the closed loop poles of OBMPC are also calculated. In this case the poles of largest module corresponding to the OBMPC of Figure 1 are: 0.7462 and $0.0882 \pm 0.1946i$. In this case the error in the critical direction decays with a time constant of 3.5 min, or the closed loop with OBMPC is about 30 times faster than DMC.

Thus, by calculating the eigenvalues and eigenvectors of the system, the two controllers can be compared without the need to simulate the system. This is of great interest as the calculation of the eigenvalues is a relative easy task and so the procedure could be used to design a controller, which will be presented in example 2.

Example 2—Using the ROSSMPC for Better Design of a DMC Controller

Suppose now that based on the closed loop information presented in example 1, it is required to improve the design of the DMC for the system of example 1 given by equation (27). Since the problem of poor performance has been diagnosed as the location of the open loop pole of the model used by DMC, the alternative to improve the performance is to use an observer faster than the real plant. As an example, consider a DMC with the following model:

$$G_M(s) = \frac{10e^{-s}}{10s + 1} \tag{28}$$

where, G_M is the model used by the controller.

Note that the model of equation (28) does not mean that the process is not exactly known. It simply means that a different model is chosen on purpose for the controller so as this is able to faster predict errors due to a disturbance in the inputs. If the process was not exactly known, the model (28) might not be the best choice.

In this example again as done by Lundström *et al.*[7] the process is assumed to be exactly known. Uncertainties in the process will be considered in sections V. and VI.

The model of equation (28) was chosen so that there would be a time constant of the observer equal to one tenth of the plant time constant. For this DMC the closed loop

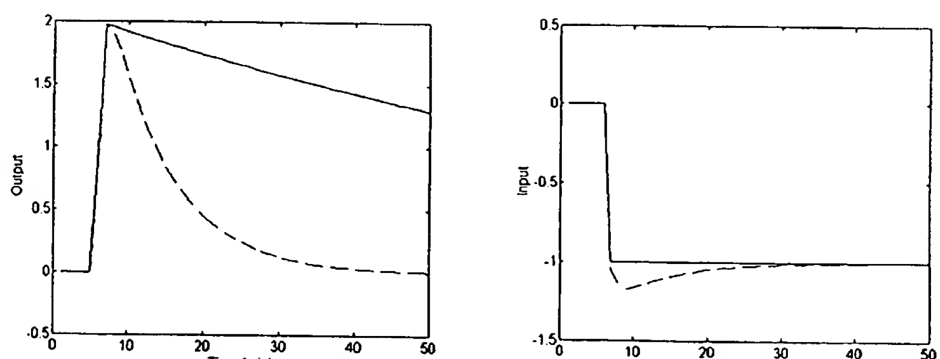

Figure 2. DMC responses for input disturbance:. (– – –) with model of equation (28) and (——) with ideal model.

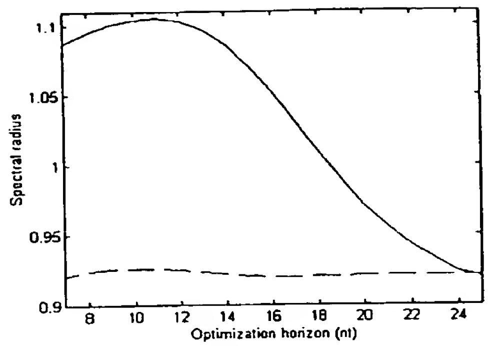

Figure 3. Example 3. Closed-loop spectral radius of conventional DMC(——) and ROSSMPC (- - - -) with the extended horizon approach.

poles become 0.8818, 0.2836 and −0.1754. The consequence of this is that the dominant time constant of the closed loop is about 8 min instead of 100 min. The responses of the system for the model of equation (28) and for the ideal case (model of equation (27)) are shown in Figure (2).

It is clear from examples 1 and 2 that the analysis of closed loop poles is feasible with the proposed state space model and becomes a useful tool in the design of MPC. The position of these poles in the complex plane is a substitute for extensive closed loop simulations.

Example 3: ROSSMPC—A More Convenient Representation

The problem considered here is a part of the Shell problem (Prett and Garcia[10]) and is given by the following model:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{1.77e^{-28s}}{60s+1} & \dfrac{5.58e^{-27s}}{50s+1} \\ \dfrac{4.42e^{-22s}}{44s+1} & \dfrac{7.20}{19s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

The large difference between the time delays of the system outputs (0 and 28) makes DMC unstable for small optimization horizons. This is so, as for small prediction horizons the error on variable y_2 will be more significant since there is no dead time for it (and variable u_2). To reduce this effect the optimization horizon must be increased and consequently the dimension of the system will be enlarged. With the ROSSMPC approach fast and slow dynamics can be balanced without increasing significantly the size of the system.

Figure 3 shows that the DMC with $T = 5$, $m = 1$, $\Lambda = 0$ and $\Gamma = I_2$, is unstable for values of n_t up to 15. In this figure is also plotted the spectral radius of the closed-loop transition matrix for the same controller, using the ROSSMPC representation extended with a single point $n_t + n_1 = 25$ and weighting the error at this point with $\Gamma_{n_t+n_1} = 10$.

Observe that for this case the controller is stable for any n_t starting from $n_t = 7$ which is the minimum value necessary to accommodate a time delay of 28 min.

Figure 4 compares the responses of the system with the conventional DMC with $n_t = 25$ and the DMC under the ROSSMPC formulation with $n_t = 7$ and $n_t + n_1 = 25$. It can be seen that the performance is almost the same. In the conventional DMC approach, the transition state matrix of the closed loop would be of dimension $120 \times 120 = 14400$,

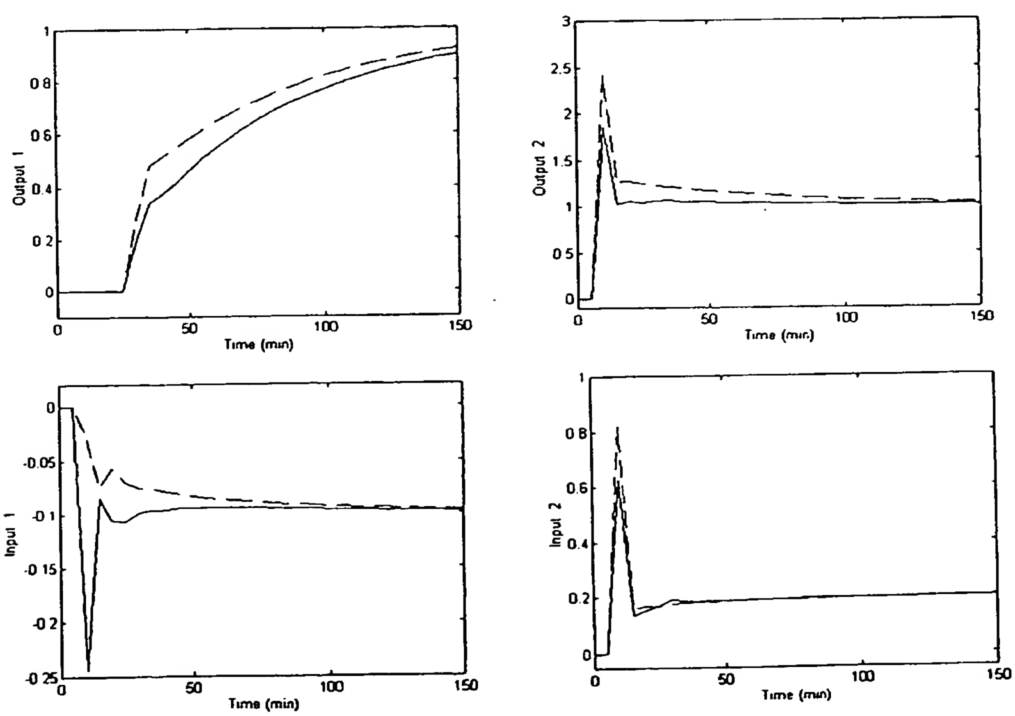

Figure 4. Example 3. DMC (- - - -) and ROSSMPC (——) response for a change on the set points of the system outputs.

while the ROSSMPC approach with the compact system representation needs a transition matrix of order $24\times24=576$. The reduction on the size of the system is enormous (96%).

In this example it was shown that with the ROSSMPC representation of a controller can be obtained without *any loss* of performance!

V. ROBUST TUNING OF MPC BY CONSTRAINING THE CLOSED LOOP POLES

In the previous section, it was shown that a poor performance of MP controllers may be explained through the location of some closed loop poles. This is e.g. the case of the DMC controller of example 1. For this example the poles were inherited from the open loop poles of the system. It might be thought that by looking at the poles of the system a controller could be designed in an adequate manner. This might be true when the model of the process is perfectly known. This is, however, not the general case. So a natural question is what will happen if there are uncertainties in the parameters of the plant model?

Besides reducing the order of the closed loop system, ROSSMPC possesses some useful features that help in the analysis of the controller. One of these features is that all the model parameters, including the time delay, appear explicitly in the state-space formulation of the system. The other advantage is that each group of parameters (gains, time constants and time delays) appears linearly in the closed loop representation of the system. This fact will be used in this section and a tool presented of robustly analysing the stability and performance of the MP controllers.

Remember that if there are errors in the identified model, these will only appear in the prediction model in the matrices M and S. So it can be assumed that the uncertainty in the matrices can be bounded as:

$$\hat{S} = S(I+\Delta_S)$$

$$\hat{M} = M(I+\Delta_M)$$

When these errors are used, the transition matrix of equation (24) can be rewritten as:

$$\begin{bmatrix} (I-RN)(M-SK_{MPC}) - RNS(I+\Delta_S)K_{MPC} & RNM(I+\Delta_M) \\ -S(I+\Delta_S)K_{MPC} & M(I+\Delta_M) \end{bmatrix} = A + E\Delta_A F \tag{29}$$

where,

$$A = \begin{bmatrix} (I-RN)(M-SK_{MPC}) - RNSK_{MPC} & RNM \\ -SK_{MPC} & M \end{bmatrix};$$

$$E = \begin{bmatrix} -RNS & RNM \\ -S & M \end{bmatrix}; \quad \Delta_A = \begin{bmatrix} \Delta_S & \\ & \Delta_M \end{bmatrix};$$

$$F = \begin{bmatrix} K_{MPC} & \\ & I \end{bmatrix} \tag{30}$$

Note that matrix A corresponds to the transition matrix of an autonomous system described by equation (31) if there were no model uncertainties.

$$x(k+1) = Ax(k) \qquad x(0) = x_0 \tag{31}$$

The eigenvalues of matrix A are supposed to be inside a specified closed contour ∂C of the complex plane. For stability reasons this contour must be inside the unit circle. Also note that an output multiplicative error is considered here as an example of how to robustly design a MPC by means of the ROSSMPC formulation. Any other model error could be used. As will be shown later, uncertainties in the parameters of the model used by ROSSMPC can be easily put in the form of equation (29).

The advantage of having the disturbed system being described by equation (29) is the possibility of using the stability radius concept of Hinrichson and Pritchard[11]. This serves as a guide in the analysis of the closed loop stability and performance of the MPC described by the ROSSMPC formulation. Their result is summarized next.

In equation (29) Δ had a particular structure, namely it was block diagonal. Let now Δ_A be a full matrix. By doing the real stability radius of the transition matrix A can be obtained for a specified contour ∂C as given in Hinrichsen and Pritchard[11] as:

$$r = \inf\left\{\bar{\sigma}(\Delta_A)\middle|\Delta_A \in R^{n_1\times n_2} \text{ and } A + E\Delta_A F \text{ has an eigenvalue at } \partial C\right\}$$

where $\bar{\sigma}(\Delta_A)$ is the largest singular value of Δ_A and where n_1 n_2 are related to the structure defined by E and F. This definition is equivalent to

$$r = \inf_{s\in\partial C} \inf\left\{\bar{\sigma}(\Delta_A)\middle|\Delta_A \in R^{n_1\times n_2} \text{ and } det(sI - A - E\Delta_A F) = 0\right\}$$

It is straightforward to show that a manipulation of the determinant leads to

$$r = \inf_{s\in\partial C} \inf\left\{\bar{\sigma}(\Delta_A)\middle|\Delta_A \in R^{n_1\times n_2} \text{ and } det\left[I - \Delta_A F(sI-A)^{-1}E\right] = 0\right\} \tag{32}$$

Equation (32) can be so used to calculate the stability radius. Its form is, however, not so simple. Here the results of Li *et al.*[12] become important, as they show a better way of how to calculate the real stability radius of real matrices, which is the case of the ROSSMPC formulation.

Let $\phi = Re\phi + jIm\phi \triangleq F(sI-A)^{-1}E$. Then Li *et al.*[12] show that for this definition the real stability radius can be calculated as:

$$r_s = \left\{\inf_{\gamma\in(0,1]} \sigma_2\begin{bmatrix} Re\phi & -\gamma Im\phi \\ Im\phi/\gamma & Re\phi \end{bmatrix}\right\}^{-1} \tag{33}$$

where σ_2 is the second largest singular value of the matrix ϕ.

$$r = \inf_{s\in\partial C}\{r_s\} \tag{34}$$

where, r_s can be considered a local stability radius associated with the point $s = x_r + x_i i$ on the contour ∂C.

Before making other considerations, it can now be shown how the above equations (33) and (34) can be used for the robust analysis of a given MP controller.

In the design stage of the MPC, the usual available information consists of the nominal model and uncertainty

bounds on its parameters obtained through extensive plant data. The uncertainty could be then limited by a maximum value, e.g. $\bar{\sigma}(\Delta_A)$. The design problem would then be the one of selecting tuning parameters such that the system would remain stable for all possible plant models. Here plant models refer to the fact that when there are uncertainties several models could apply to the process depending on the actual operating conditions.

When the uncertainties in the plant model can be limited the design procedure is given as:

select the tuning parameters so that $r \geq \bar{\sigma}(\Delta_A)$ (P1)

There are, however, situations where it is not known how large the uncertainties can be. For these cases the design procedure must be slightly modified. Here tuning parameters must be selected that will enforce stability for the largest possible set of plant models, which can be viewed as the following procedure:

$$\max_{\text{tuning parameters}} r \qquad \text{(P2)}$$

Thus the stability radius can play a major role in the robust stability design of a controller. Nonetheless some conservatism can be introduced if the stability radius is not adequately calculated. This will e.g. be the case when no structural information is taken into account.

This is what happens with equation (33). There no structural information about the model error is considered besides that given by matrices E and F. Consequently, if equation (33) is used to calculate the stability radius, a conservative result is obtained as Δ_A is considered to be a full matrix. So it is desirable to calculate the stability radius in a less conservative manner. Note that under the ROSSMPC formulation it was shown that if an output multiplicative error is considered Δ_A becomes diagonal. For this case it will be shown next how to modify equation (33).

For any diagonal matrix Ω, the following identity is valid:

$$\begin{aligned} det(I - \Delta_A\phi) &= det(I - \Omega\Delta_A\phi\Omega^{-1}) \\ &= det(I - \Delta_A\Omega\phi\Omega^{-1}) \end{aligned}$$

Making use of this identity, equation (33) can be rewritten as:

$$r_s = \left\{ \inf_{\Omega} \left\{ \inf_{\gamma \in (0,1]} \sigma_2 \begin{bmatrix} Re\tilde{\phi} & -\gamma Im\tilde{\phi} \\ Im\tilde{\phi}/\gamma & Re\tilde{\phi} \end{bmatrix} \right\} \right\}^{-1} \qquad (35)$$

where, $\tilde{\phi} = \Omega\phi\Omega^{-1}$.

For the ROSSMPC, equation (35) can be used instead of equation (33) which gives better results. It remains now to show how errors in the parameters of the models can be expressed in the form of equation (30) so that equation (35) might be used. This will be done in section VI for the cases where there is model mismatch in the static gains and in the dead-time of the process.

VI. APPLYING THE ROBUST DESIGN PROCEDURE TO SOME EXAMPLES

In section V, the way the stability radius can be calculated was represented for the special model error structure presented in equations (29) and (30). It was also claimed that the ROSSMPC formulation allows the representation of the real system in this form. It was also shown that the robust tuning procedure of a MP controller under the ROSSMPC formulation can be stated as procedure (P1) or (P2). Here the aforementioned topics are illustrated in examples 4 and 5.

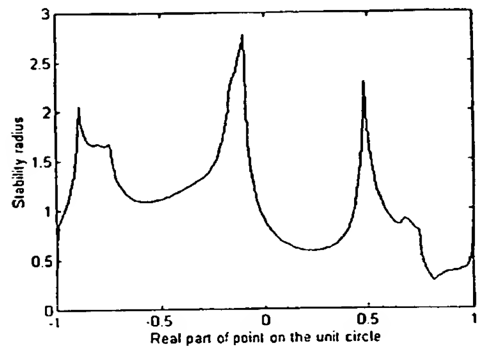

Figure 5. Example 4. Local stability radius for ROSSMPC.

In example 4 the following topics are covered:

- there will be uncertainties in the static gains of the model: it will be shown how to obtain the structure of equation (30)
- some tuning parameters of the ROSSMPC controller will be selected and how the stability radius is used to analyse the robustness of the closed loop system shown. First the stability case is considered and later it is shown how the performance can be analysed.
- once the analysis tool is described procedure (P2) is applied to show how the parameters of the ROSSMPC controller can be tuned to achieve robust performance and stability.

In example 5 the following aspects are considered:

- there will be uncertainties in the time delay of the system: it will be shown how to obtain the structure of equation (30)
- it is illustrated how the QDMC controller used by Zafiriou[18] and Lee & Yu[19] can be better analysed
- once the behavior of the QDMC controller is understood, the effect of the mismatch in the time delay in the stability and performance of the system is shown.

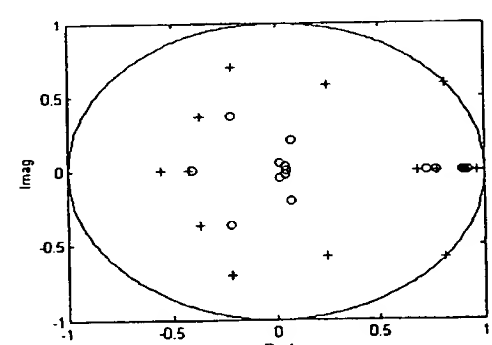

Figure 6. Example 4. Closed loop poles (+) for a plant with $k_1 = 0.2711$, $k_2 = -0.2711$ and with ideal model (o).

Example 4—Robust Analysis of the Shell Problem—Uncertainties in the Static Gains of the Model

The process of example 3 is again considered, except that here it is assumed that there are uncertainties in the static gains as proposed by Prett and Garcia[10]:

$$G(s) = \begin{bmatrix} \dfrac{1.77(1+k_1)e^{-28s}}{60s+1} & \dfrac{5.58(1+k_2)e^{-27s}}{50s+1} \\ \dfrac{4.42(1+k_1)e^{-22s}}{44s+1} & \dfrac{7.20(1+k_2)}{19s+1} \end{bmatrix}$$

For this special case it is noted that $\Delta_M = 0$ and Δ_S is a diagonal matrix given by:

$$\Delta_S = \begin{bmatrix} k_1 & \\ & k_2 \end{bmatrix}$$

As a consequence, in the ROSSMPC structure E and F can be further reduced to:

$$E = \begin{bmatrix} -RNS \\ -S \end{bmatrix}; \quad F = \begin{bmatrix} K_{MPC} & 0 \end{bmatrix}$$

In the analysis that follows ROSSMPC will be applied to this system with the following tuning parameters:

$$T = 5, \; m = 2, \; n_r = 7, \; n_1 = 25,$$
$$\Gamma_{n_1} = 10 \text{ and } \Lambda = 0 \tag{36}$$

and the maximum values of the gain uncertainties k_1 and k_2 will be calculated where the system still remains stable.

Suppose further that any point of the unit circle can be represented by $x_r + x_i i$. Figure 5 shows the plot of r_s calculated with equation (35) as a function of x_r.

From Figure 5 it is clear that the optimization problem defined in equation (34) is not convex as the function that relates r_s and x_r shows several local minimum points. The global minimum has to be selected by inspection, but this is not a major problem since the search is unidirectional and bounded. Then, it is concluded that for the selected controller, the stability radius of the system is $r = 0.2711$, corresponding to the point on the unit circle defined by $x_r = 0.810$. This means that with the adopted tuning parameters, the system under ROSSMPC control is stable as long as the uncertainties obey the following condition: $\bar{\sigma}(\Delta_S) \le 0.2711$.

These results can be interpreted by examining the poles of the closed loop system. Figure 6 shows the dominant poles of the closed loop system. There two situations are considered:

- no model uncertainties
- model uncertainties defined by $k_1 = 0.2711$ and $k_2 = -0.2711$.

It is observed that for the latter case there are poles at $0.810 \pm 0.587i$ which are exactly on the unit circle. Any further increase in the model uncertainty will push these poles outside the unit circle and the system with the selected ROSSMPC parameters will become unstable.

Up to here only stability considerations were discussed. Another very important feature of any controller is its performance and moreover its robust performance. Here, the way the stability radius can be used to enforce robust performance is discussed.

Performance criteria can be established by means of the suitable location of the closed loop poles on the complex plane. The ideal situation would be the case where all the poles of the closed loop system were at the origin of the complex plane. In this case the system output would exactly follow the set-point. As the ideal situation is not usually feasible, there must be a performance criterion. Here it is considered that a process has good performance if none of the following situations occur:

- underdamped oscillatory systems with a small damping ratio. This corresponds to the case where the response of the system will oscillate for a long time before it finally

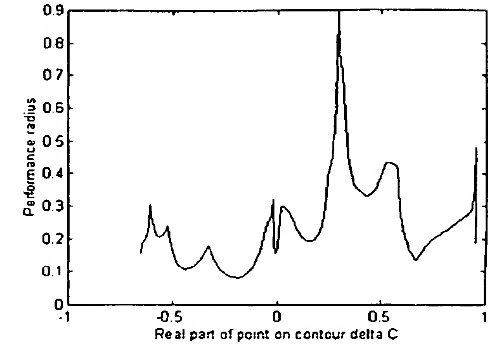

Figure 8. Example 4. Performance radius for contour ∂C_1.

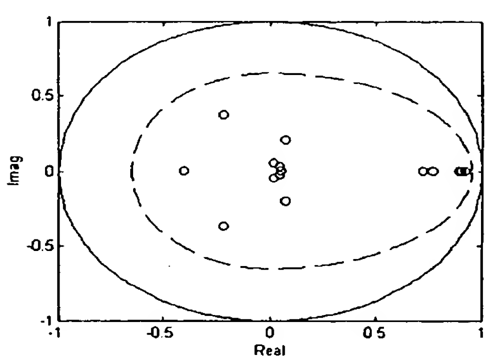

Figure 7. Example 4. Closed loop poles for ideal model (o) and contour ∂C (– – – –) that define a performance region on the complex plane.

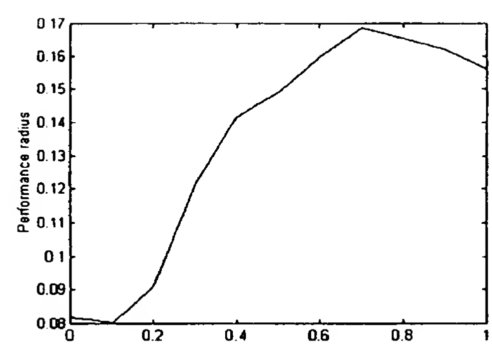

Figure 9. Example 4. Relation between the performance radius and input weight Λ.

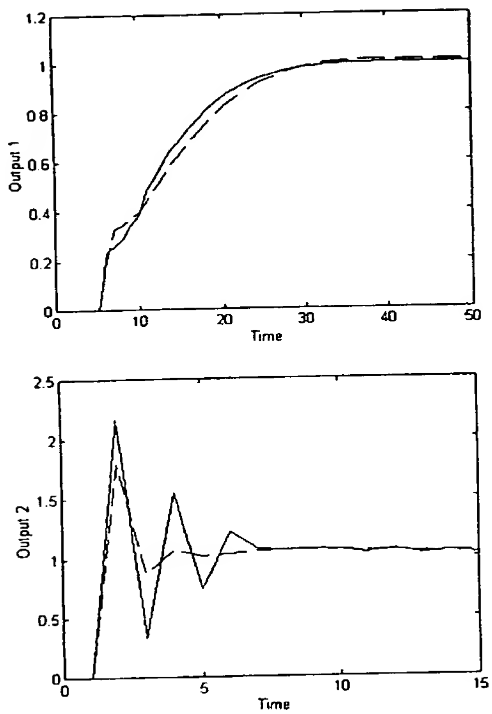

Figure 10. Example 4. System outputs for a set-point change with $\Lambda = 0$ (——) and $\Lambda = 0.7$ (– – –).

stabilizes. This kind of response is associated with poles with large negative real part or large imaginary part. Consequently this kind of response can be prevented by further constraining the poles of the closed loop into a suitable sub-region of the unit circle.

• sluggish or overdamped systems with a large dominant time constant. This kind of response is related to the presence of closed loop poles on the real axis close to (1,0).

The robust performance design procedure could thus be viewed as the one of selecting a contour on the complex plane and of finding tuning parameters, that would produce a stability radius larger than the model uncertainty.

Next the way to evaluate performance requirements for this example will be discussed. The contour represented in Figure 7, named ∂C_1, are selected as our performance criterion. It will limit a certain region of the complex plane, which will have the characteristics described next.

On the left half side of the complex plane, contour ∂C_1 is a circle with radius $r_1 = 0.65$ and on the right half side it is an ellipsis with axes $r_1 = 0.65$ and $r_2 = 0.95$. The region limited by these contours enforces the following characteristics on the system:

• the closed loop system will have a damping ratio of at least 2.5 for any plant model. Also the closed loop will never have a time constant larger than 78 min.

For simplicity of notation, the stability radius related to this contour will be called the performance radius. Figure 8 shows the local performance radius of ROSSMPC with the parameters given as in equation (36), as a function of x_r for contour ∂C_1. By inspecting this curve it is observed that the performance radius has the value $r_s = 0.0818$ at $x_r = -0.1885$. This means that for any uncertainty Δ_S for which $\bar{\sigma}(\Delta_S) \leq 0.0818$, all the closed loop poles of the system will stay inside the region defined by contour ∂C_1.

In equation (36) Λ was considered to be equal zero. For this case it was seen that the system is to be stable. Except that the bound on $\bar{\sigma}(\Delta_S)$ will not be large. So a natural question would be how to better select Λ so that a larger limit on the uncertainty is obtained. The procedure that must be used to answer this question is the one given in (P2). That is what will be done next.

For simplicity it is assumed that the two inputs have the same weight so that Λ will have the same diagonal entries. So whenever a value is assumed for Λ, it means that its diagonal elements have this value.

Figure 9 shows the effect of input weight Λ on the performance radius. It is observed that the performance radius can be maximized by properly selecting the input weight. For this example the performance radius has a maximum value at $\Lambda = 0.7$, where the performance radius is close to $r_s = 0.17$.

This result means the following: 'the maximum error in the static gains must obey the following condition $\bar{\sigma}(\Delta_S) = 0.17$ if the system is to attain the performance criterion ∂C_1.' Note that the controller with $\Lambda = 0$, does not attend the adopted performance criteria as $\bar{\sigma}(\Delta_S) = 0.17 > 0.0818$. This does not mean that it will be unstable, only that the performance will not be acceptable for larger errors in the static gains. To better illustrate this point consider next that the uncertainties in the static gains are defined by $k_1 = k_2 = 0.17$. Figure 10 shows the response of the system with model mismatch for a set-point change, the system being controlled by the ROSSMPC controller tuned with $\Lambda = 0$ and $\Lambda = 0.7$. The responses of output y_1 are quite similar for both input weights, but for output y_2 the system response does not have the required damping ratio when $\Lambda = 0$.

Example 5—Effect of Uncertainties in the Time Delay

Consider the SISO system used by Zafirou[18] and Lee and Yu[19] to study the loss of robustness of MPC in the presence of uncertainty in the time delay, when there is hard constraint in the system output.

$$G(s) = \frac{1}{s+1} e^{-0.15s}$$

In this case the step response of the nominal system is:

$$s(t) = 1 - 1.e^{(t-0.15)}$$

While the true plant model is represented by:

$$\hat{s}(t) = 1 - 1.e^{(t-0.15-\Delta\theta)}$$

where $\Delta\theta$ is the time delay mismatch between the true plant model and the model used by ROSSMPC. With this assumption the plant step response can be written as follows:

$$\hat{S} = S + V\Delta\bar{\theta}$$

where

$$V = \left[-e^{-(2T-0.15)} - e^{-(3T-0.15)} \ldots - e^{-((n_s+1)T-0.15)} \; 0 \; 0\right]^T$$

$$\Delta\bar{\theta} = e^{-\Delta\theta} - 1$$

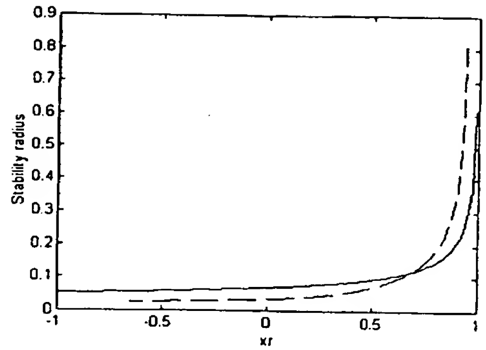

Figure 11. Example 5. Stability radius (————) and performance radius for contour ∂C_1 (– – –).

Applying these definitions in equation (29) we can obtain the following relations:

$$E\Delta_A F = \begin{bmatrix} -RNV & RNW_1 \\ -V & W_1 \end{bmatrix} \begin{bmatrix} \Delta\bar{\theta} & 0 \\ 0 & \Delta\bar{\theta} \end{bmatrix} \begin{bmatrix} K_{MPC} & 0 \\ 0 & W_2 \end{bmatrix} \tag{37}$$

where

$$W_1 = \begin{bmatrix} \underbrace{0 \quad 0 \quad \dots \quad 0}_{n_t - 1} & e^{-[(n_t+1)T-0.15]} & 0 & 0 \end{bmatrix}^T$$

and

$$W_2 = [0 \quad \dots \quad 0 \quad 1]$$

It is easy to see that matrix E in equation (37) is rank deficient and consequently matrix $\phi = F(sI - A)^{-1}E$ is singular for any point s on the chosen contour ∂C. As $det(I - \Delta_A\phi) = 0$ it can be shown that the stability radius associated with point s can be calculated by:

$$r_s = |\Delta\bar{\theta}| = |1/(\phi_{1,1} + \phi_{2,2})| \tag{38}$$

where $\phi_{1,1}$ and $\phi_{2,2}$ are the diagonal elements of matrix ϕ.

Zafiriou[18] and Lee and Yu[19] used the QDMC as the controller for this system. They have also shown by simulation that when $\Delta\theta = 0.15$, the closed loop is unstable if a hard constraint in the system output becomes active. Here this will be analysed by means of ROSSMPC. The nature of critical values of the delay mismatch for both

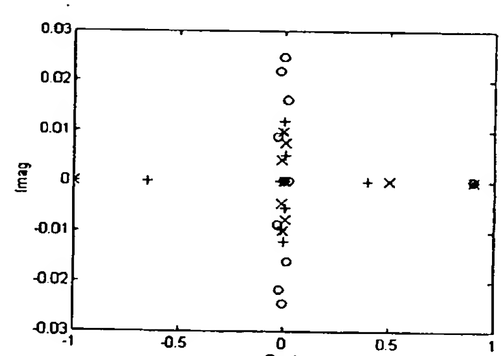

Figure 12. Example 5. Closed loop poles for $\Delta\theta = 0$ (o); $\Delta\theta = -0.0513$ (x) and $\Delta\theta = -0.0260$ (+).

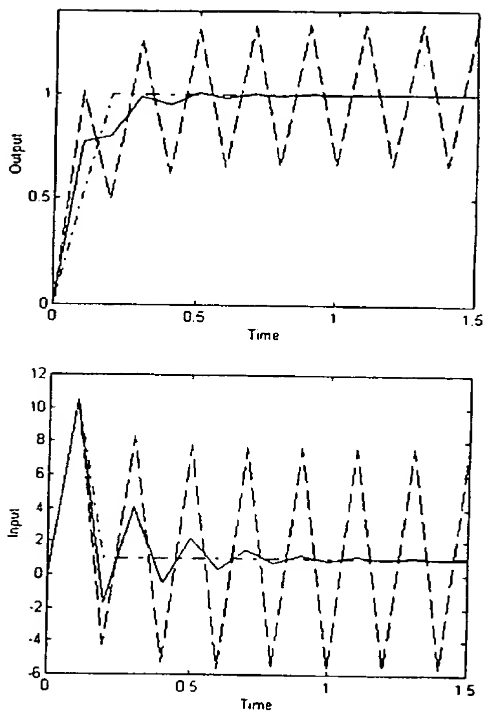

Figure 13. Example 5. System input and output for a set-point change: $\Delta\theta = 0$ (– · – · –); $\Delta\theta = -0.0513$ (– – –) and $\Delta\theta = -0.0260$ (————).

robust stability and robust performance will be studied.

Robust performance will be defined in the same way as in the previous example, i.e., the system will have a good performance if the closed loop poles are within the contour ∂C_1 defined in example 4.

The authors cited used the QDMC controller with hard constraint in the output. So first it must be ascertained how to formulate this controller using the ROSSMPC. The situation corresponding to having in the QDMC the constraints in the output active, is the one of the dead-beat tuning of ROSSMPC. With the least squares control objective function, this is obtained when the input suppression factor equals zero and the control horizon equals the number of prediction instants beyond the time delay in the optimization horizon. If there are no constraints in the input, it is possible to obtain control actions, that will eliminate the error between the predicted trajectory of the system output and its set-point.

So, consider now the ROSSMPC with the following tuning parameters:

$$T = 0.1, \quad \Lambda = 0, \quad m = 5 \text{ and } n_t = m + 2$$

Results are independent of the control horizon m, as long as the optimization horizon is equal to the control horizon plus the time delay. The stability radius of the system calculated by equation (38) is shown in Figure 11 as a function of the real part of the point s on the unit circle. The minimum of r_s is equal to 0.0488 and is obtained for $x_r = -1$. This means that the controller can be unstable if

$$|\Delta\bar{\theta}| = |e^{\Delta\theta} - 1| \geq 0.0488$$

The critical case corresponds to $\Delta\theta = -0.0513$ where

one of the closed loop poles lies at $(-1,0)$, as shown in Figure 12. This explains why QDMC is unstable when $\Delta\theta = 0.15$, as shown by Zafiriou[18]. The value of the uncertainty used by that author in his simulations was much larger than the allowable value that can be obtained by inspecting the stability radius of the system.

Analogously to the procedure followed in example 4, a performance criterion can also be specified for this system, by constraining the poles of the closed loop system into a suitable closed contour of the complex plane. Adopting the performance contour ∂C_1, the critical mismatch in the time delay becomes $\Delta\theta = -0.0260$. The dominant pole of the closed loop system lies at $(-0.65,0)$. This is also shown in Figures 11 and 12.

To better clarify these results, Figure 13 shows the response of the system for a set-point change for the following cases:

- no model mismatch
- a model mismatch given by $\Delta\theta = -0.0513$
- a model mismatch given by $\Delta\theta = -0.0260$

This example shows that ROSSMPC has a significant advantage over the other MPC representations when robust stability and performance of the system is the issue for time delay uncertainty. The model representation of ROSSMPC allows a simple and straight approach to solving the problem.

VII. CONCLUSIONS

In this paper a state space representation has been presented for the class of model predictive controllers, named ROSSMPC. This representation can be used for any existing MP controller. For example, ROSSMPC was applied for the DMC, QDMC and OBMPC controllers. So ROSSMPC can be also regarded as a way of describing existing MP controllers. This is of great interest as several industrial applications make use of the DMC controller. While preserving the properties of the existing controllers on the one side, ROSSMPC permits us to better explore and comprehend the controller chosen. This is so as the ROSSMPC is based on a step-response model and preserves the parameters of the model in an explicit manner in the state space representation. Moreover the number of states is reduced. As a consequence the eigenvalues or poles of the closed-loop transfer function can be evaluated and a better comprehension of the behaviour of the closed-loop system is acquired without the need of extensive simulation. This also makes the procedure of tuning the controller simpler. The parameters of the controller can be so chosen as to make the poles of the transfer function lie in desired regions of the complex plane. For the ROSSMPC representation, the prediction of the behavior of the system in the future can be arbitrarily made and there is also no need of using equally spaced time instants. For processes where there is the necessity to use large control horizons, this is of great advantage. Stability can be ensured and at the same time the number of states remains small. ROSSMPC can be used both as an analysis tool and as a way of modelling and designing a controller. For the first case a given MP controller can be better designed.

For example, in this paper it was shown how the model used by the DMC controller can be modified in order to enhance its performance. That is, it was shown that sometimes introducing known errors in the model of the process can make the controller respond faster to disturbances. This is an interesting result and shows that the model itself can be regarded as another parameter that can be selected to enhance the performance of control systems. As the ROSSMPC model possesses fewer parameters than other MP models, it can be used as a controller and so the computer work of calculating control actions is reduced. This is important for the cases where there is interest in having one or very few controllers for the whole unit. ROSSMPC has a structure that is also suitable for the use of robust analysis tools.

In this paper it was shown how the stability radius can be used for allowing the controller to work well and maintain robust stability and performance. This was, for example, accomplished by limiting the closed loop poles inside suitable regions of the complex plane.

NOMENCLATURE

A	transition matrix for the closed loop with nominal model
c	parameter of the continuous step response function
C	matrix of parameters of the step response function
d	coefficient for updating the prediction function parameters, equation (12)
D	matrix of coefficients for updating the prediction function parameters
E, F	matrices defining the structure of the uncertainty, equation (30)
G	process transfer function
f	function in the continuous step response, equation (3)
K_{MPC}	MPC feedback gain
m	control horizon
M	state matrix in the prediction model, equation (4)
$\hat{M}$	state matrix of the plant real model
n_h	step or impulse response model horizon
n_r	optimization horizon
n_t	first part of the prediction horizon of ROSSMPC
$\hat{n}_t$	prediction horizon of the plant in the reduced order model
n_u	number of inputs
n_y	number of outputs
N	matrix in the error correction equation (5)
$P, \bar{P}$	parameters of the output prediction function, equation (8)
R	disturbance correction coefficient for DMC, equation (5)
s	step response coefficient of the model in MPC
$\hat{s}$	step response coefficient of the plant
S	step response coefficient matrix of the prediction model, equation (4)
$\hat{S}$	plant step response coefficient matrix, equation (22)
$\bar{S}$	dynamic matrix of MPC
t	time
T	sampling period
u	manipulated input
υ	Eigenvector of the closed loop transition matrix
y	system output

Greek symbols

Γ	output weighting matrix, equation (19)
Λ	input weighting matrix, equation (19)
θ	time delay
ψ	matrix in the output prediction, equation (10)
σ	singular value
μ	structured singular value

Abbreviations

DMC	dynamic matrix controller
MAC	model algorithmic control
MP	model predictive
MPC	model predictive controller
MPHC	model heuristic predictive control
OBMPC	observer based model predictive control
QDMC	quadratic dynamic matrix control
ROSSMPC	reduced order state space model predictive controller

REFERENCES

1. Li, S., Lim K. Y. and Fisher, D. G., 1989, A state space formulation for model predictive control, *AIChE J*, 35: 241–249.
2. Richalet, J., Raoult, A., Testud, J. L. and Papan, J., 1978, Model predictive control: application to industrial processes, *Automatica*, 14: 413–428.
3. Cutler, C. R. and Ramaker, B. L., 1980, Dynamic matrix control—a computer control algorithm, *Proc Joint Automatic Control Conf, San Francisco, CA*, Paper WP5-B.
4. Prett, D. M. and Gillette, R. D., 1979, Optimization and constrained multivariable control of a catalytic cracking unit, *AIChE National Meeting, Houston*.
5. Rouhani, R. and Mehra, R. K., 1982, Model algorithmic control (MAC): Basic theoretical properties, *Automatica*, 18: 401–410.
6. Guettler, S. J., Glasgow, D. T. and Scott, D. E., 1993, A novel approach to multivariable control, *48th Annual Texas A&M Instrumentation Symposium*.
7. Lundstrom, P., Lee, J. H., Morari, M. and Skogestad, S., 1995, Limitations of dynamic matrix control, *Comp Chem Eng*, 19(4): 409–421.
8. Lee, J. H., Morari, M. and Garcia, C. E., 1994, State space interpretation of model predictive control, *Automatica*, 30: 707–717.
9. Hovd, M., Lee, J. H. and Morari, M., 1993, Truncated step response model predictive control, *J Process Control*, 3(2): 67–73.
10. Prett, D. M. and Garcia, C. E., 1988, *Fundamental Process Control*, (Butterworths, Stoneham MA).
11. Hinrichsen, D. and Pritchard, A. J., 1986, Stability radius for structured perturbations and the algebraic Riccati equation, *Syst Control Lett*, 8: 105–113.
12. Li Qiu, Bernhardsson, B., Rantzer, A., Davison, E. J., Young, P. M. and Doyle, J. C., 1995, A formula for computation of the real stability radius, *Automatica*, 31(6): 879–890.
13. Bailay, J. K., 1995, Process identification using finite impulse response models, *J Process Control*, 5(2): 77–84.
14. Ljung, L., 1987, *System Identification: Theory for the User*. (Prentice-Hall, Englewood Cliffs, NJ).
15. Åström, K. J. and Wittenmark, B., 1990, *Computer Controlled Systems. Theory and Practice*, (Prentice Hall, Englewood Cliffs, NJ, USA).
16. Rawlings, J. B. and Muske, K. R., 1993, The stability of constrained multivariable receding horizon control, *Trans Autom Cont*, 38(10): 1512–16.
17. Morari, M. and Lee, J. H., 1991, Model predictive control: The good, the bad and the ugly, *Proc CPC IV, San Padre Island*, 419–444.
18. Zafiriou, E., 1991, The robustness of model predictive controllers, *Proc. CPC-IV, San Padre Island*.
19. Lee, J. H and Yu, Z. H., 1994, Tuning of model predictive controllers for robust performance, *Comp Chem Eng*, 18(1): 15–37.
20. Oliveira, N. M. C. and Biegler, L., 1994, Constraint handling and stability properties of model predictive control, *AIChEJ*, 40(7): 1138–1155.

ACKNOWLEDGEMENT

Support for this research was provided by Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) under grants 93/3184-0 and 96/4482-2.

ADDRESS

Correspondence concerning this paper should be addressed to Dr D. Odloak, Chemical Engineering Department, University of São Paulo, C.P. 61548-05424-970 São Paulo, SP, Brazil. (E-mail: odloak@usp.br).

The manuscript was received 8 July 1996 and accepted for publication after revision 21 May 1997.